Device Class Power Management Reference Specification

Communications Device Class

Draft proposal
v0.0

Table of Contents

Revision History

Revision	Date	Comments
0.0	3/25/96	Initial proposal for consideration

Scope

This specification defines the behavior of Communications devices as it relates to power management, and, specifically, to the four device power states defined for the OnNow Architecture. This specification applies to Communications ports and modems. It is intended that communications device vendors and system makers will be able to design consistent power-manageable products, and that OS vendors will be able to implement an appropriate communications device power management policy based on the contents of this specification.

General Device Power Management Considerations

In the OnNow architecture, power management of individual devices is the responsibility of a policy owner in the Operating System, generally a class-specific driver. This policy-owner will implement a power conservation policy that is appropriate for devices in its class. The policy will operate in conjunction with a global system power policy implemented in the operating system (i.e. is the system Working or Sleeping?). In general, the device-class power conservation policy strives to reduce power consumption while the system is Working by transitioning amongst various available power states according to device usage. Since the policy-owner in the Operating System has very specific knowledge of when a device is in use, or potentially in use, there is no need for hardware timers or such to determine when to make these transitions. Similarly, this level of understanding of device usage makes it possible to use fewer device power states. Generally, intermediate states attempt to draw a compromise between latency and consumption due to the uncertainty of actual device usage. With the increased knowledge in the OS, crisp decisions can be made about whether the device is needed at all. With this ability to turn devices off more frequently, the benefit of having intermediate states diminishes.

The policy-owner also determines what class-specific events can cause the system to transition from Sleeping to Working, and enables this functionality based on application or user requests. Note that the definition of the wake-up events that each class supports will influence the system's global power policy in terms of the level of power conservation the Sleeping state can attain while still meeting wake-up latency requirements set by applications or the user.

In the OnNow architecture, bus drivers also implement power policy for their bus class (e.g. PCI, USB, etc..). In general, the Bus driver has responsibility for tracking the device power states of all devices on its bus, and transitioning the Bus itself to only those power states that are consistent with those of its devices. This means that the Bus state can be no lower than the highest state of one of it's devices. However, enabled wake-up events can affect this as well. For example if a particular device is in the D2 state and set to wake-up the system, and the bus can only forward wake-up requests while in the D1 state, then the Bus must remain in the D1 state even if all devices are in a lower state.

Device power state transitions are explicitly commanded by the driver. The actual power control is handled through methods defined in the various bus standards (e.g. ATA Standby command, USB Suspend, etc..). In some cases, these bus-standard mechanisms are not available and device-specific mechanisms must be used. Note that the explicit command for entering the D3 state may be the removal of power.

The following definitions apply to devices of all classes:

- **D0**: Device is on and running. It is receiving full power from the system, and is delivering full functionality to the user.
- **D1**: Class-specific low-power state (defined below) in which device context may or may not be lost. Buses in D1 cannot do anything to the bus which would force devices on that bus to loose context.
- **D2**: Class-specific low-power state (defined below) in which device context may or may not be lost. Attains greater power savings than D1. Buses in D2 may cause devices on that bus to loose some context (e.g. the bus reduces power supplied to the bus). Devices in D2 must be prepared for the bus to be in D2 (or higher).
- **D3**: Device is off and not running. Device context is lost. Power may be removed from the device. Any device context lost must be restored by the device driver when returning the device to the D0 state.

Communications Port Power State Definitions

D0

- Line drivers: On (powered)
- UART Context: Preserved

D1

This state is not defined for Communications Ports. Use the D3 state instead.

D2

This state is not defined for Communications Ports. Use the D3 state instead.

D3 (Power may be removed)

- Line drivers: Off (Unpowered; Outputs isolated from devices attached to the port)
- UART Context: Lost

Latency to return to D0: Less than TBD

Communications Port Power Conservation Policy

Present State	Next State	Cause
D3	D0	• Power-on reset • COM port opened by an application
D0	D3	• COM port closed • System enters sleeping state while wake is disabled on this device • System enters sleeping state while wake is enabled on this device and the device is capable of waking from state D3

Communications Port Wake-up Events

Asserting the Ring Indicate line will cause the communications port to assert a wake-up event.

Modem Power State Definitions

D0

- Phone interface: On; may be on or off hook
- Speaker: On
- Controller Context: Preserved

D1

- Phone interface: Not powered by host; On Hook
- Speaker: Off
- Controller Context: Preserved

D2

- This state is not defined for Modems. Use the D3 state instead.

D3 (Power may be removed)

- Phone interface: Not powered by host; On Hook
- Speaker: Off
- Controller Context: Lost

Modem Power Conservation Policy

Present State	Next State	Cause
D3	D0	Modem opened by an application
D0, D1	D3	• Modem closed • System enters sleeping state while wake is disabled on this modem • System enters sleeping state while wake is enabled on this modem and the modem is capable of waking from state D3
D0	D1	• Modem put in answer mode • System enters sleeping state while wake is enabled on this modem and the modem is capable of waking from state D1 but not from state D3
D1	D0	• Application requests dial • Phone rings while modem is in answer mode

Modem Wake-up Events

An incoming phone call will cause the modem to assert a wake-up event.

Minimum Communications Device Power Capabilities

TBD